

Inhaltsverzeichnis

1	Verzeichnisse und Referenzen	1
1.1	Referenzieren	2
1.2	hyperref	3
2	Mathematik	3
2.1	Fonts und Text in Gleichungen	3
2.2	Einheiten	4
2.3	Beschriftung von Gleichungsteilen	5
2.4	Theorem-Umgebung	5
3	Eigene Befehle und Umgebungen	6
4	Modularisierung und Bedingungen	7
5	Titelseite anpassen	7
6	Index und Abkürzungen	9
6.1	Index	9
6.2	Abkürzungen	10

1 Verzeichnisse und Referenzen

Wie schon in früheren Vorträgen erwähnt, bietet \LaTeX automatisch erstellte Verzeichnisse an. Das wichtigste Beispiel ist das Inhaltsverzeichnis, welches per `\tableofcontents` ausgegeben werden kann. Für Gleitobjekte werden ebenfalls Verzeichnisse erstellt. Diese können mit `\listoffigures` und `\listoftables` ausgegeben werden. Generell sind Objekte, die in ein solches Verzeichnis aufgenommen werden, referenzierbar. Um sie zu referenzieren, müssen sie mit `\label{Markierung}` markiert werden. Die Markierung sollte möglichst mit einem Präfix versehen sein, welches der in Tabelle 1 gezeigten Konvention entspricht. Dies ist für manche Befehle notwendig, um den Typ einer Referenz zu bestimmen, zudem hilft es, die Übersicht über die Markierungen zu behalten.

Sowohl bei Verzeichnissen, als auch bei Referenzen ist zu beachten, dass `(pdf)latex` mehrfach aufgerufen werden muss: Im ersten Durchlauf findet \LaTeX alle Markierungen

Präfix	Typ	Präfix	Typ
ch:	Kapitel	sec:	Abschnitt
subsec:	Unterabschnitt	fig:	Abbildung
tab:	Tabelle	eq:	Gleichung
lst:	Code Listing	app:	Anhang

Tabelle 1: Konventionsgemäße Präfixe

und Objekte, die in ein Verzeichnis eingetragen werden sollen und speichert sie in einer entsprechenden Datei, z.B. `datei.toc` für das Inhaltsverzeichnis und `datei.aux` für die Referenzen. Im nächsten Durchlauf werden diese Dateien verwendet, um das Verzeichnis oder die Referenz korrekt aufzubauen. Außerdem werden die Hilfsdateien erneut erzeugt. Ein dritter Aufruf stellt nun sicher, dass z.B. die Seitenzahlen im Inhaltsverzeichnis korrekt sind. Diese können falsch werden, wenn sich Seiten, z.B. durch ein größer gewordenes Abbildungsverzeichnis, verschoben haben.

Möchte man die Nummerierung unterdrücken, so kann man meist ein „*“ anfügen, z.B. `\section*{Abschnitt}`. Da der Abschnitt nun keine Nummer mehr bekommt, wird er aber auch nicht ins Inhaltsverzeichnis eingefügt. Man kann ihn aber von Hand mit dem Befehl `\addcontentsline{Verzeichnis}{Ebene}{Text}` einfügen. Ebene ist z.B. `section`, das Inhaltsverzeichnis heißt `toc`. Falls Überschriften sehr lang sind, kann mit einem optionalen Argument ein Kurztitel vergeben werden. Dieser wird dann anstelle der Überschrift in das Inhaltsverzeichnis übernommen. Beim `\caption[Kurztitel]{Titel}`-Befehl funktioniert dies analog. Hier ist es besonders hilfreich, da beschreibende Abbildungsbeschriftungen meist nicht geeignet sind, in einem Abbildungsverzeichnis aufzutauchen. Will man festlegen, bis zu welcher Ebene das Inhaltsverzeichnis ausgegeben wird, so kann dies mit `\setcounter{tocdepth}{Tiefe}` erreichen. Dabei wird eigentlich nur der Zähler `tocdepth` verändert, aus dem L^AT_EX ausliest, welche Tiefe verwendet werden soll. Der `\setcounter`-Befehl ist also auch für alle anderen Zähler nutzbar.

1.1 Referenzieren

Wurden nun Objekte mit `\label{Markierung}` markiert, soll üblicherweise auch auf sie zugegriffen werden. Dazu kann der `\ref{Markierung}`-Befehl verwendet werden. Dieser gibt aber nur die Nummer der Markierung aus, der Typ muss also ergänzt werden. Um eine Abbildung zu referenzieren, muss also ein solcher Code verwendet werden:

```
In Abbildung~\ref{fig:Abbildung} ist etwas zu sehen.
```

Zwischen den Typ und den Befehl `\ref` sollte ein nicht umbrechendes Leerzeichen (~) gesetzt werden, da sonst eventuell zwischen dem Typ und der Nummer umgebrochen wird.

Analog kann der `\pageref{Markierung}`-Befehl verwendet werden. Er gibt die Seite, auf der die Markierung auftaucht, aus. Für Gleichungen bietet `amsmath \eqref{Markierung}`. Dieser Befehl gibt neben der Gleichungsnummer auch Klammern aus. So liefert der Quelltext

```
\begin{equation}
  a^2+b^2=c^2\label{eq:pythagoras}
\end{equation}
```

```
Der Satz des Pythagoras steht in \eqref{eq:pythagoras}.
```

die folgende Ausgabe:

$$a^2 + b^2 = c^2 \tag{1}$$

Der Satz des Pythagoras steht in (1).

Die Referenzen sollten am Besten direkt in der Beschriftung des entsprechenden Objekts eingetragen werden. Damit ist sichergestellt, dass das richtige Objekt markiert wird.

1.2 hyperref

Das `hyperref`-Paket erweitert die Referenzfähigkeiten von L^AT_EX noch weiter: Es wandelt Referenzen automatisch in Links im Dokument um. Außerdem bietet es den Befehl `\url{URL}` an, um externe URLs, z.B. auf Websites oder Mailadressen, zu setzen. In der Standardeinstellung zeichnet `hyperref` allerdings bunte Kästen um Links. Um dies zu verhindern, kann die Option `colorlinks=true` gesetzt werden. Nun werden Links eingefärbt. Da das Farbschema sehr bunt ist, bietet es sich an, außerdem die Optionen `linkcolor`, `citecolor`, `filecolor` und `urlcolor` zu nutzen. Mit `urlcolor=blue` werden beispielsweise URLs blau eingefärbt.

Um Tipparbeit zu sparen, kann auch der Befehl `\autoref{Markierung}` verwendet werden. Dieser gibt z.B. bei Abbildungen dann nicht nur die Nummer (z.B. „1“) sondern auch die Bezeichnung (z.B. „Abbildung 1“) aus.

2 Mathematik

2.1 Fonts und Text in Gleichungen

Falls in Gleichungen spezielle Fonts verwendet werden sollen, bietet sich das Paket `amsfonts` an. Dieses stellt z.B. `\mathbb{N}` (\mathbb{N}), `\mathcal{N}` (\mathcal{N}) und `\mathfrak{N}` (\mathfrak{N}) bereit. Bereits in L^AT_EX enthalten ist `\mathrm{N}` (N). Diese wurde bereits in früheren Vorträgen bei Integralen genutzt:

$$\backslash[\int x\mathrm{d}x\backslash]$$
$$\int x dx$$

Neben speziellen Fonts taucht auch oft normaler Text in Gleichungen auf. Die Mathe-Umgebung würde den Text einfach ohne Leerzeichen und kursiv darstellen. Daher kann `\text{Text}` verwendet werden, welches den Text wie im übrigen Dokument setzt. Wenn Text in einem Gleichungsblock (`align` o.ä.) zwischen Gleichungen eingefügt werden soll, kann `\intertext{Text}` genutzt werden.

Das Ohmsche Gesetz

$$U = R \cdot I \quad (\text{U: Spannung, I: Strom, R: Widerstand}) \quad (2)$$

kann nach dem Widerstand

$$R = \frac{U}{I} \quad (3)$$

oder dem Strom

$$I = \frac{U}{R} \tag{4}$$

umgestellt werden.

Um dies zu erhalten, wurde dieser Quelltext verwendet:

Das Ohmsche Gesetz

```
\begin{align}
U&=R\cdot I\text{(U: Spannung, I: Strom, R: Widerstand)} \\
&\intertext{kann nach dem Widerstand}
R&=\frac{U}{I} \\
&\intertext{oder dem Strom}
I&=\frac{U}{R}
\end{align}
```

umgestellt werden.

2.2 Einheiten

Um Einheiten sauber in L^AT_EX zu setzen, sollte das Paket `siunitx` verwendet werden. Dieses bietet zwei wichtige Befehle: `\SI{Wert}{Einheit}` setzt einen Wert mit der entsprechenden Einheit, `\si{Einheit}` gibt die Einheit aus. Dabei wird die Einheit geschrieben. Um mm auszugeben wird also Beispielsweise `\si{\milli\meter}` geschrieben. Es kann auch `\per` oder `\square` verwendet werden, wenn durch Einheiten geteilt wird oder sie als Quadrat vorkommen. Dieser Quelltext

```
$W$=\SI{10}{\kilogram\meter\per\square\second} \\
\si{\tera\hertz}-Strahlung mit $\lambda$=\SI{100}{\micro\meter}$
```

demonstriert, dass `siunitx` für einheitlichen Satz sorgt, egal ob die Einheit im Text oder in einer Mathe-Umgebung steht:

$W=10 \text{ kg m s}^{-2}$

THz-Strahlung mit $\lambda = 100 \mu\text{m}$

Da die Funktionen des Pakets sehr vielfältig sind, ist im Folgenden nur ein kurzer Überblick über die wichtigsten Fähigkeiten gegeben. Unter <http://mirrors.ctan.org/macros/latex/contrib/siunitx/siunitx.pdf> findet sich die sehr umfangreiche Dokumentation zu diesem Paket.

Neben der Einheiten Ausgabe kann `siunitx` auch Zahlen vereinheitlichen. Es ersetzt beispielsweise in der Standardeinstellung „.“ als Dezimaltrenner durch „.“. Dies kann natürlich über eine Option (`output-decimal-marker = {,}`) umgestellt werden. Außerdem werden Schreibweisen wie $1e4 \text{ m}$ oder 10000 m zu $1 \times 10^4 \text{ m}$ vereinheitlicht. Das

Ausgabeformat kann natürlich auch eingestellt werden, dazu sei aber auf die Dokumentation verwiesen. Mit `\DeclareSInit{Einheit}{Inhalt}` können auch neue Einheiten definiert werden. Dieser Befehl muss vor dem `\begin{document}` stehen.

Zudem bietet das Paket z.B. die Möglichkeit, in Tabellen eine Spalte am Dezimaltrenner auszurichten.

2.3 Beschriftung von Gleichungsteilen

Um Gleichungsteile zu beschriften bietet L^AT_EX vier Befehle an. `\overbrace{Text}^{Oben}` und `\underbrace{Text}_{Unten}` setzen Text mit einer horizontalen geschweiften Klammer über beziehungsweise unter den Gleichungsteil. Möchte man dies ohne Klammer, kann man `\overset{Oben}{Gleichungsteil}` oder `\underset{Unten}{Gleichungsteil}` verwenden:

```
\[\overbrace{a+b}^2+\underbrace{c+d}_3\overset{\text{gleich}}{=}5\]
```

$$\overbrace{a+b}^2 + \underbrace{c+d}_3 \overset{\text{gleich}}{=} 5$$

2.4 Theorem-Umgebung

A_MS bietet das Paket `amsthm` an, mit dem Theoreme gesetzt werden können. Es liefert die `proof`-Umgebung mit, um Beweise zu setzen. Diese werden dabei automatisch nummeriert. Außerdem können Stile mit dem Befehl `\theoremstyle{Stil}` genutzt werden, um das Aussehen anzupassen. Will man andere Theoreme, z.B. Sätze oder Lemmata, setzen, kann man sich mit `\newtheorem{Name}[numName]{Markierung}[Ebene]` neue Umgebungen definieren. Dabei ist `Name` der Name der Umgebung und `numName` der Name der Umgebung, mit der diese gemeinsam nummeriert werden soll. Damit können z.B. Sätze und Axiome mit der gleichen fortlaufenden Nummerierung gesetzt werden. Wenn als erstes ein Satz definiert wird, erhält dieser die Nummer Eins. Das nächste Axiom dann die Nummer Zwei und da darauf folgende Axiom die Nummer Drei. `Markierung` ist der Name, der im Dokument ausgegeben wird und mit `Ebene` kann festgelegt werden, in welcher Strukturebene die Nummerierung stattfindet. Wird hier beispielsweise `section` gesetzt, beginnt die Nummerierung in jedem Abschnitt neu. Dieser Quellcode

```
\newtheorem{satz}{Satz}
\begin{satz}
  \LaTeX{} ist f"ur Formelsatz geeignet.
\end{satz}
\begin{satz}
  \LaTeX{} ist nicht wirklich kompliziert.
\end{satz}
```

führt also zu dieser Ausgabe:

Satz 1. *L^AT_EX ist für Formelsatz geeignet.*

Satz 2. *L^AT_EX ist nicht wirklich kompliziert.*

Weiteres zu diesem Paket findet ihr in der Dokumentation: <https://www.ctan.org/pkg/amsthm>

3 Eigene Befehle und Umgebungen

L^AT_EX ist eigentlich „nur“ eine Makro-Sammlung für T_EX. Prinzipiell können natürlich die gleichen, in T_EX vorhandenen Erweiterungsmöglichkeiten, genutzt werden. L^AT_EX bietet aber einfach zu benutzende Befehle an, um sich eigene Befehle (Makros) oder Umgebungen zu definieren. Ein eigener Befehl kann mittels `\newcommand{\Befehl}[Anzahl der Argumente]{Inhalt}` definiert werden. Analog dazu gibt es den `\renewcommand`-Befehl, der einen vorhandenen Befehl überschreibt. `\providecommand` funktioniert wie `\newcommand`, allerdings prüft es vorher, ob ein Befehl schon vorhanden ist und tut nichts, wenn dies der Fall ist. `\newcommand` würde dagegen in diesem Fall einen Fehler melden.

In einem selbst definierten Befehl können auch Argumente verwendet werden. Diese werden mit #1, #2, etc. angesprochen. Um sich einen Befehl zu definieren, der Abbildungen referenziert und dabei jeweils „Abbildung“ vor die Nummer schreibt, könnte man also einen solchen Code benutzen:

```
\newcommand{\figref}[1]{Abbildung~\ref{#1}}
```

Will man nun stattdessen „Abb.“ vor die Abbildungsnummer schreiben, kann man den Befehl so überschreiben:

```
\renewcommand{\figref}[1]{Abb.~\ref{#1}}
```

Für Umgebungen gibt es den Befehl `\newenvironment{Name}[Anzahl der Argumente]{Vorne}{Hinten}`. Dieser funktioniert ähnlich zu `\newcommand`, aber statt nur eines Inhaltsfeldes kann unter `Vorne` Code angegeben werden, der an der Position des `\begin`-Befehls ausgeführt wird. Analog wird der Inhalt von `Hinten` am Ende der Umgebung ausgeführt. Um dieses Beispiel zu erhalten,

Anfang Inhalt Ende

ist folgender Quelltext notwendig:

```
\newenvironment{gross}
  {\vspace{2mm}\hrule Anfang\Large}
  {\normalsize Ende\\\hrule}
\begin{gross}
  Inhalt
\end{gross}
```

4 Modularisierung und Bedingungen

Wie die meisten Programmiersprachen bietet auch L^AT_EX die Möglichkeit, externe Dateien einzubinden. Damit können Dokumente über mehrere Dateien verteilt werden. Beispielsweise kann eine Datei pro Kapitel in einer Dissertation verwendet werden. Am Einfachsten ist der Befehl `\input{Datei}`, welcher praktisch den Quelltext von `Datei` an der Stelle des `\input`-Befehls im Dokument einfügt. Entsprechend kann dieser Befehl überall im Dokument stehen.

Im Gegensatz dazu darf `\include{Datei}` nur nach dem `\begin{document}` auftauchen. Außerdem fügt es `\clearpage` nach dem eingebundenen Quelltext ein. Es bietet aber mit dem Befehl `\includeonly{Datei1,Datei2,...}` die Möglichkeit, nur bestimmte `\include`-Befehle auszuführen. Dies kann hilfreich sein, wenn man zum Korrigieren nur einzelne Kapitel einer großen Arbeit kompilieren möchte. In diesem Beispiel werden also nur `kap1` und `kap3` wirklich eingebunden:

```
\input{header}
\includeonly{kap1,kap3}
\begin{document}
  \include{kap1}
  \include{kap2} %wird nicht eingebunden
  \include{kap3}
  ...
```

Insbesondere in diesem Zusammenhang ist auch das Paket `ifthen` hilfreich. Es bietet ein einfach zu nutzendes `if/then/else`-Konstrukt in Form des Befehls `\ifthenelse{Bedingung (if)}{Inhalt (then)}{Alternative (else)}`. Dabei können `Inhalt` und `Alternative` natürlich auch leer bleiben. Der folgende Quelltext prüft, ob das Makro `draft` den Inhalt `true` besitzt. Wenn dies nicht der Fall ist, wird `anhang` eingebunden und ein Abbildungsverzeichnis erstellt:

```
\ifthenelse{\not\equal{\draft}{true}}
{ %kein draft
  \input{anhang}
  \listoffigures
}{ %draft, nichts tun
}
```

Um dies praktisch zu Nutzen, ist es hilfreich, dass man `(pdf)latex` auch direkt Quelltext angeben kann. Der (Kommandozeilen-)Befehl `pdflatex '\newcommand{\draft}[true]\input{beispiel}'` kompiliert `beispiel.tex` mit dem Makro `\draft`, das „`true`“ als Inhalt hat.

5 Titelseite anpassen

Insbesondere bei Abschlussarbeiten gibt es oft Vorgaben, wie die Titelseite aussehen soll. Dies ist in L^AT_EX mit der `titlepage`-Umgebung relativ einfach umzusetzen. Diese Um-

(a) Schriftgrößen		(b) Wichtige Schrifttypen	
Befehl	Schriftgröße	Befehl	Schrifttyp
<code>\tiny</code>	<code>\tiny</code>	<code>\textrm</code>	Roman
<code>\scriptsize</code>	<code>\scriptsize</code>	<code>\textbf</code>	Fett
<code>\footnotesize</code>	<code>\footnotesize</code>	<code>\textsc</code>	KAPITÄLCHEN
<code>\small</code>	<code>\small</code>	<code>\textsf</code>	Serifenlos
<code>\normalsize</code>	<code>\normalsize</code>	<code>\textit</code>	<i>kursiv</i>
<code>\large</code>	<code>\large</code>		
<code>\Large</code>	<code>\Large</code>		
<code>\LARGE</code>	<code>\LARGE</code>		
<code>\huge</code>	<code>\huge</code>		
<code>\Huge</code>	<code>\Huge</code>		

Tabelle 2: Schriftgrößen und -Typen in L^AT_EX

gebung erzeugt eine leere Seite und setzt den Seitenzähler zurück. Innerhalb dieser Seite kann dann beliebig positioniert werden. Wenn weiterhin mit `\author{Autor}` etc. Informationen zum Dokument definiert werden sollen, muss vor der Titelseite `\makeatletter` aufgerufen werden, um entsprechende Makros mit dem Namen `\@author` etc. zu erzeugen. Im folgenden Beispiel wurde die Schrift serifenlos (`\sf`) gesetzt und einzelne Elemente in verschiedenen Größen ausgegeben. Außerdem wurde ein Logo eingefügt. Dies wird hier *nicht* in eine `figure`-Umgebung gesetzt, da die Positionierung nicht von L^AT_EX übernommen werden soll. Im Beispiel werden Schrifttypen und Schriftgrößen verwendet. Eine Übersicht zu diesen findet sich in Tabelle 2.

```

\documentclass{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage{graphicx}
\author{Jan-Martin R"amer}
\date{18.05.2015}
\title{Ein Testdokument}
\begin{document}
  \begin{titlepage}
    \makeatletter\sf
    \begin{center}
      \quad\vskip 5cm
      {\Huge \@title}\\[1cm]
      \rule{.75\linewidth}{.5mm}\\[1cm]
      {\Large \@author}\\[2cm]
      {\large \@date}\\[4cm]
      \includegraphics[width=5cm]{U}
    \end{center}
  \end{titlepage}
\end{document}

```

```
\end{center}  
\end{titlepage}  
\end{document}
```

Das Ergebnis findet sich in Abbildung 1.

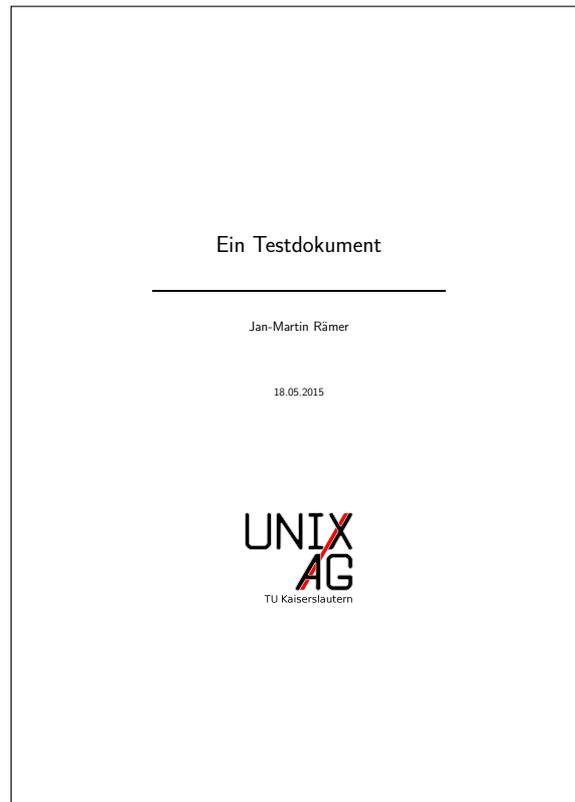


Abbildung 1: Veränderte Titelseite

6 Index und Abkürzungen

6.1 Index

Um in L^AT_EX einen Index zu erzeugen, kann das Paket `makeidx` zusammen mit dem (Kommandozeilen-)Befehl `makeindex` verwendet werden. `makeindex` baut dabei den Index auf, weshalb entsprechend wieder mehrfach kompiliert werden muss.

Um Einträge einzufügen, gibt es den Befehl `\index{Eintrag}`. Dieser verwendet einige Sonderzeichen: Mit einem „!“ werden Untereinträge getrennt: `\index{Eintrag! Untereintrag}`. Ein „@“ kann verwendet werden, um die Sortierung festzulegen, beispielsweise um β unter „b“ einzusortieren: `\index{b@$\beta}`. Will man auf einen anderen Eintrag verweisen, geht dies mittels „|see“ wie in `\index{pdflatex|see{latex}}`. Da die hier genannten Zeichen nun nicht direkt verwendet werden können, wird „+“

als Maskierungszeichen verwendet. Für ein Ausrufezeichen muss also ein solcher Code verwendet werden: `\index{wichtig+!}`.

Damit der Index erstellt wird, muss vor dem `\begin{document}` `\makeindex` aufgerufen werden. Zur Ausgabe des Index' wird der Befehl `\printindex` verwendet. Nachdem der Index mit `makeindex Dateiname` in der Kommandozeile aktualisiert wurde, muss er mit einem weiteren (pdf)latex-Lauf eingebunden werden. Dabei ist es sinnvoll, den Dateinamen ohne Endung anzugeben.

6.2 Abkürzungen

Gerade in technischen Texten tauchen oft viele Abkürzungen auf, die dem Leser nicht unbedingt alle bekannt sind. Zudem entspricht es einem guten Stil, Abkürzungen beim ersten Auftreten im Text auszuschreiben. Das Paket `acronym` unterstützt dies. In der `acronym`-Umgebung können mit dem Befehl `\acro{Akronym}[Kurzname]{Name}` Einträge in das Abkürzungsverzeichnis hinzugefügt werden. An der Stelle dieser Umgebung wird dann auch das Abkürzungsverzeichnis ausgegeben. Alternativ können Einträge auch mittels `\defacro{Akronym}[Kurzname]{Name}` eingefügt werden. Der Befehl `\newacro{Akronym}[Kurzname]{Name}` fügt zwar auch eine Abkürzung ein, erstellt aber keinen Eintrag im Abkürzungsverzeichnis. Um Pluralformen einzufügen, gibt es analog zu den genannten Funktionen `\acroplural`, `\defacroplural` und `\newacroplural`. Wenn keine Pluralform definiert ist, fügt `acronym` einfach ein „s“ an.

Im Regelfall werden Abkürzungen mit `\ac{Akronym}` genutzt. Dieser Befehl gibt bei der ersten Verwendung die Langform aus und fügt die Abkürzung in Klammern an. Bei weiteren Verwendungen wird dann nur die Abkürzung ausgegeben. Außerdem wird die Abkürzung als Link auf das Abkürzungsverzeichnis gesetzt, sofern das Paket `hyperref` nach `acronym` eingebunden wird. Soll nach einem bestimmten Punkt, z.B. nach dem Abstract, wieder zunächst die Langform ausgegeben werden, kann `\acresetall` verwendet werden. Wenn gezielt eine Form ausgegeben werden soll, kann mit `\acf` die Langform mit Abkürzung in Klammern, mit `\acs` die Kurzform und mit `\acl` die Langform ohne angefügte Abkürzung ausgegeben werden. Ein `p` am Ende dieser Befehle, z.B. `\acp` führt dazu, dass die Pluralform verwendet wird. Ein Beispiel ist hier gezeigt, in Abbildung 2 sieht man die Ausgabe:

```
\begin{acronym}
  \acro{TDS}[THz-TDS]{THz Time Domain Spectroscopy}
  \acro{OAPM}{Off-Axis Parabolic Mirror}
  \acro{DUT}{Device Under Test}
  \acroplural{DUT}[DUTs]{Devices Under Test}
\end{acronym}
\begin{tabular}{|l|l|l|}\hline
Erste Verwendung & Zweite Verw. & Plural\\\hline
\ac{TDS} & \ac{TDS} & \acp{TDS}\\\hline
\ac{OAPM} & \ac{OAPM} & \acp{OAPM}\\\hline
\ac{DUT} & \ac{DUT} & \acp{DUT}\\\hline
\end{tabular}][2mm]
```

Grundlagen 4

L^AT_EX-Kurs der Unix-AG

Langer Plural von DUT: `\aclp{DUT}`

THz-TDS THz Time Domain Spectroscopy

OAPM Off-Axis Parabolic Mirror

DUT Device Under Test

Erste Verwendung	Zweite Verw.	Plural
THz Time Domain Spectroscopy (THz-TDS)	THz-TDS	THz-TDSs
Off-Axis Parabolic Mirror (OAPM)	OAPM	OAPMs
Device Under Test (DUT)	DUT	DUTs

Langer Plural von DUT: Devices Under Test

Abbildung 2: Beispiel zu acronym